



Documentatie proiect

Prelucrare Grafica

Autor: Pescaru Silviu-Mihaita

Grupa: 30235

Profesor indrumator: Sabou Adrian

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

18 Ianuarie 2023

Cuprins

1 Prezentarea temei	2
2 Scenariul	2
2.1 Descrierea scenei si a obiectelor	2
2.2 Functionalitati	2
3 Detalii de implementare	2
3.1 Functii si algoritmi	2
3.2 Modelul grafic	2
3.3 Structuri de date	3
3.4 Ierarhia de clase	3
4 Prezentarea interfetei grafice utilizator	3
5 Manual de utilizare	6
6 Concluzii si dezvoltari ulterioare	7
7 Referinte	7

1 Prezentarea temei

Tema proiectului: Realizarea unei scene 3D in OpenGL

2 Scenariul

2.1 Descrierea scenei si a obiectelor

Scena 3D prezinta obiecte texturate ce formeaza 2 satuce inconjurate de munti din toate partile si despartite de apa. Tot in scena se pot vedea 2 mori de vant si foarte multi copaci.

2.2 Functionalitati

Avem functionalitati mobile(miscarea in scena prin tastatura si miscarea camerei din mouse), posibilitatea de a alege care dintre cele 2 surse de lumina sa fie folosite(directionala sau punctiforma sau ambele) si schimbarea modurilor de vizualizare(solid, wireframe, poligonal, smooth si depth map).

3 Detalii de implementare

3.1 Functii si algoritmi

Probabil cele mai importante functii din aceasta aplicatie grafica sunt:

- `void keyboardCallback(GLFWwindow* window, int key, int scancode, int action, int mode)`

Este functia care ne permite sa alegem sursele de lumina si modurile de vizualizare.

- `void mouseCallback(GLFWwindow* window, double xpos, double ypos)`

Cu aceasta functie putem folosi mouse-ul pe post de orientare in scena.

- `void processMovement()`

Aceasta este alta functie foarte importanta deoarece este folosita pentru a putea realiza translatia in scena dar si pentru a putea roti lumina globala.

- `void initModels()`

Fara aceasta functie nu am putea avea obiecte in scena.

- `void initShaders()`

Fara aceasta functie placa grafica nu ar sti ce si cum sa coloreze un pixel.

- `void initUniforms()`

Este functia unde avem toata partea matematica necesara pentru tot ce tine de perspectiva camerei, matricea normalelor si mai ales pozitia si culoarea luminilor din scena 3D.

- `void renderScene()`

Toate functiile de mai sus ar fi de prios fara incepe practic "desenarea" tuturor obiectelor, luminilor si umbrelor.

3.2 Modelul grafic

Pentru a putea pune cap la cap toata scena am folosit aplicatia Blender unde am importat toate obiectele necesare scenei mele. Tot in aceasta aplicatie am realizat si texturarea tuturor obiectelor, modelarea muntelui si determinarea pozitiilor anumitor obiecte din scena.

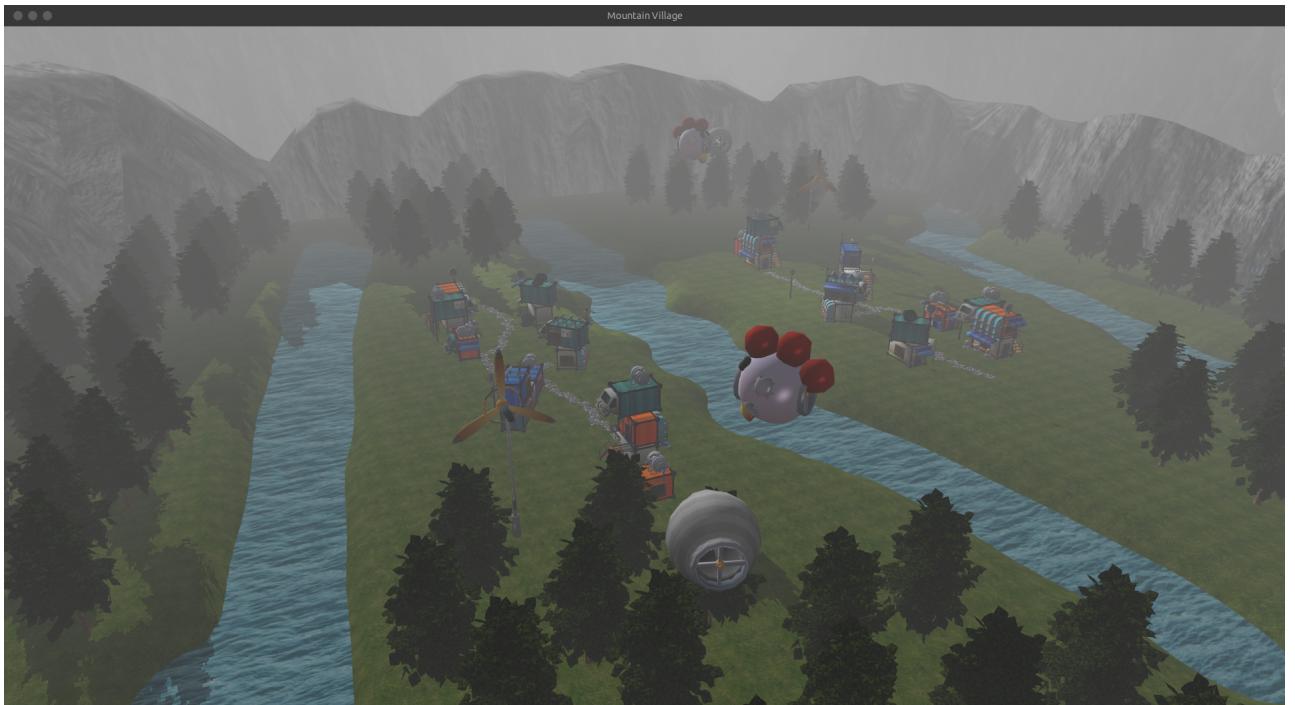


Figura 1: Mountain Villages

3.3 Structuri de date

Se folosesc structuri de date precum vectorii glm pentru OpenGL, tipurile GLuint, GLfloat, Model si Shaders, si tipuri obisnuite din C++, ca bool, int si float.

3.4 Ierarhia de clase

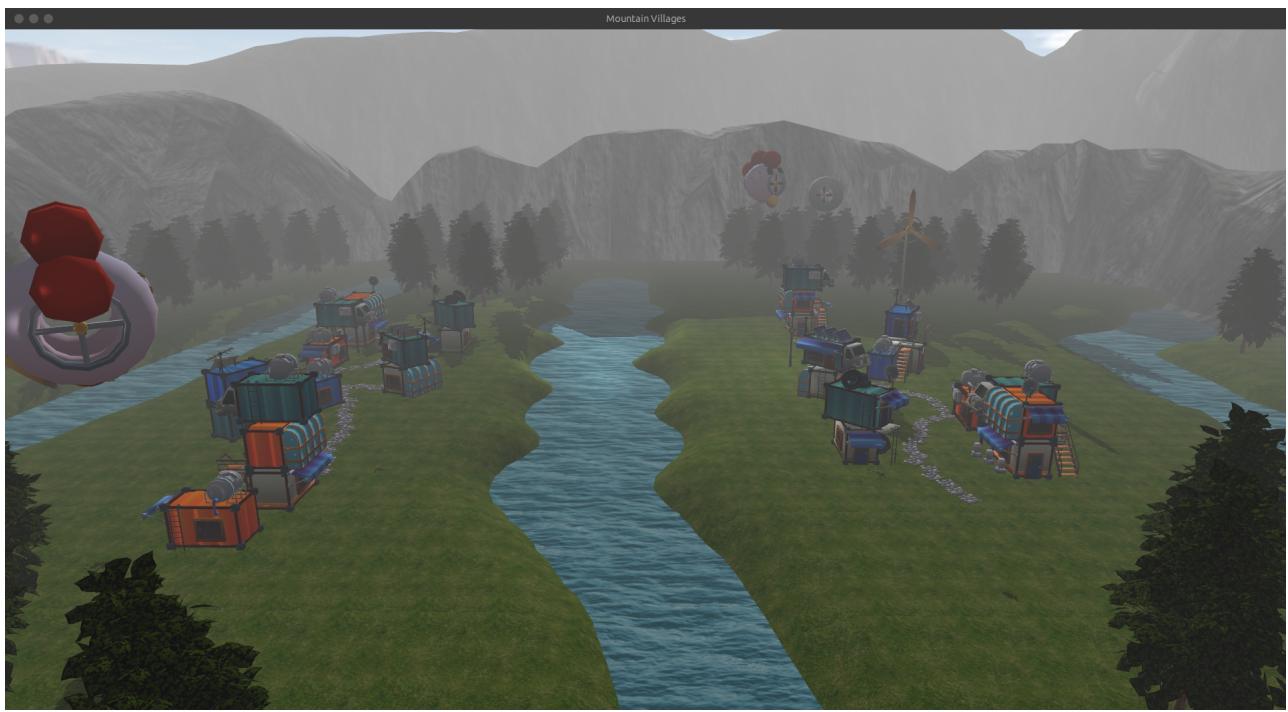
Clasa main este de sine statatoare. Restul claselor din proiect vin insotite si de headere.

- **main.cpp**: Punctul de pornire al programului. Contine initializari ale componentelor si creeaza instantia Window.
- **Camera.cpp** si **Camera.hpp**: Are ca scop gestionarea perspectivei si pozitionarii camerei. Cu ajutorul ei miscam si controlam camera in scena.
- **Mesh.cpp** si **Mesh.hpp**: Se ocupa cu triunghiurile care definesc geometria obiectelor tridimensionale.
- **Model3D.cpp** si **Model3D.hpp**: Gestionarea unui obiect 3D format din mai multe mesh-uri.
- **Shader.cpp** si **Shader.hpp**: Gestionarea shaderele OpenGL. Ofera functionalitati precum incarcarea si compilarea acestora.
- **Skybox.cpp** si **Skybox.hpp**: Gestionarea si afiseaza skybox-ului, un cub al carui fete sunt imagini cu cer si pamant, pentru a crea fundalul scenei 3D.
- **stb_image.cpp** si **stb_image.hpp**
- **tiny_obj_loader.cpp** si **tiny_obj_loader.hpp**
- **Window.cpp** si **Window.hpp**: Reprezinta fereastra aplicatiei in care se afla scena.

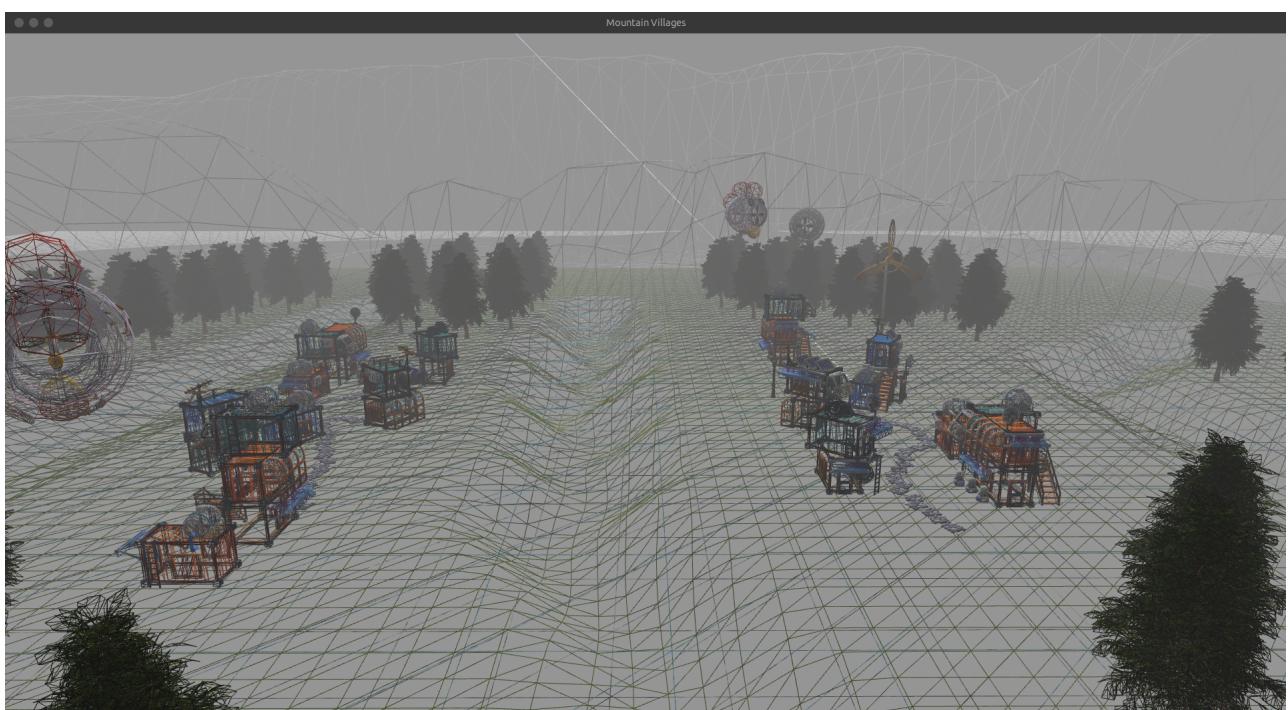
4 Prezentarea interfetei grafice utilizator

Moduri de vizualizare:

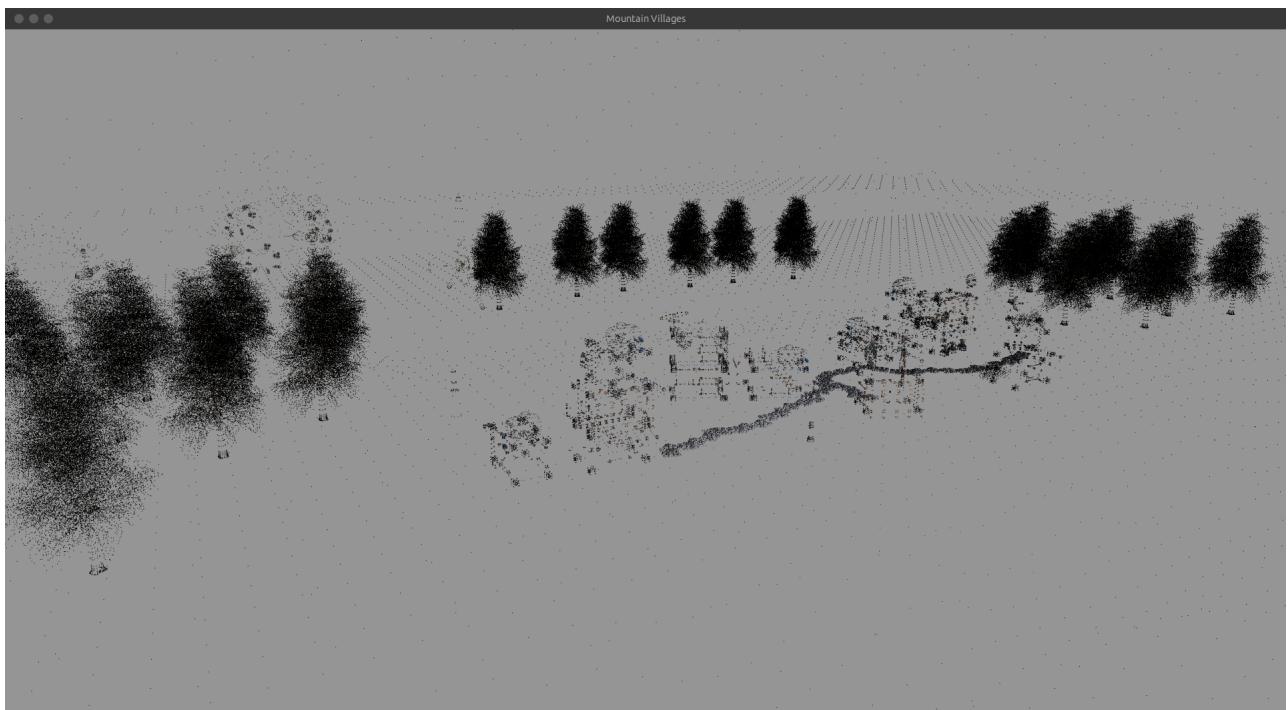
- Solid



- Wireframe



- Point



- Cu ceata



- Cu umbre



- Lumina punctiforma



5 Manual de utilizare

Utilizatorul are la indemana urmatoarele functionalitati:

- Tasta W: deplasare in fata
- Tasta S: deplasare in spate

- Tasta A: deplasare la stanga
- Tasta D: deplasare la dreapta
- Tasta Q: rotire lumina directionala la stanga
- Tasta E: rotire lumina directionala la dreapta
- Tasta R: toggle ceata
- Tasta T: toggle lumina punctiforma
- Tasta F: modul solid
- Tasta G: modul wireframe
- Tasta H: modul punct
- Tasta J: modul smooth

6 Concluzii si dezvoltari ulterioare

Proiectul realizat în OpenGL, care include funcționalități de bază precum mișcarea cu mouse-ul și tastatura, generarea luminilor și umbrelor, a fost o experiență educativă valoroasă. Implementarea unei scene 3D interactive a contribuit la consolidarea cunoștințelor teoretice și la dobândirea abilităților practice în grafica pe calculator.

7 Referinte

1. <https://learnopengl.com/>
2. <https://sketchfab.com/>
3. <https://free3d.com/>
4. <https://turbosquid.com/>
5. Tutoriale Blender: Tutoriale Blender Constantin Nandra
6. Materialele și coduri surse din laborator